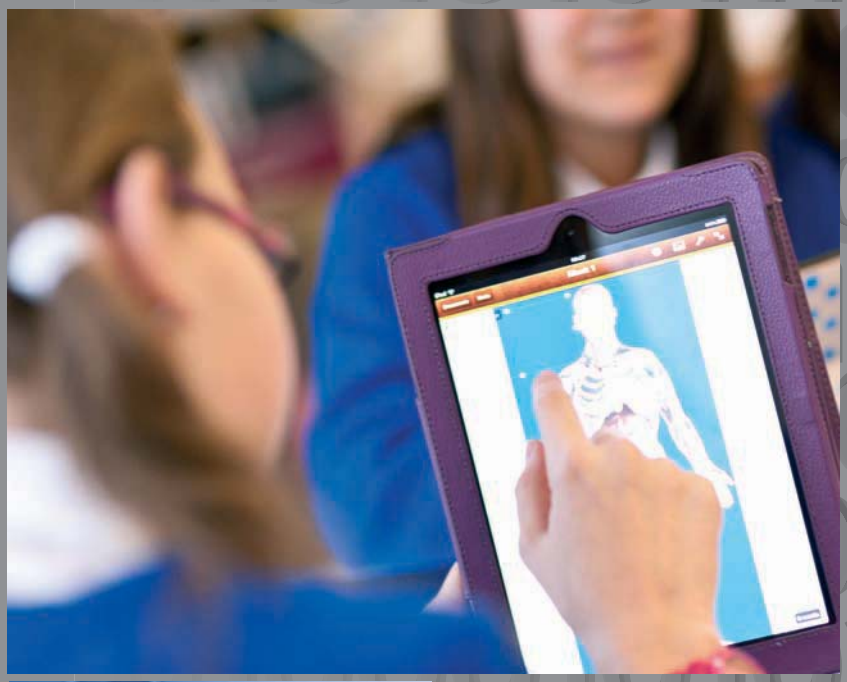
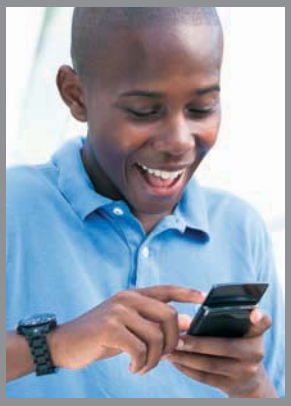
GET WITH THE PROGRAM!

If we're keen to raise contributing citizens rather than simple consumers, a rethink of the way we approach ICT in schools is vital, insists Paul Clarke

For a few years now there has been a growing number of teachers who have been working to put computer science and computational thinking back into ICT being taught in schools. For evidence of this growth one only has to look at the Computing At School (CAS) Group (www.computingatschool.org.uk) – a gathering of people, some like me from industry but mostly teachers, who share this passion. Between January 2010 and January 2012, membership of CAS rocketed from 200 to more than 1,000.



Before looking at why computer science is important and what it means for teachers, it's worth noting some very significant events from the past twelve months that show this evolution of thinking about ICT is becoming more high-profile:



- Eric Schmidt from Google challenged the government, arguing that the country that invented the computer was 'throwing away your great computer heritage' by failing to teach programming in schools
- David Willetts announced a "Behind the Screen" pilot to encourage computer programming in schools
- OCR GCSE Computing GCSE moved from pilot phase to a fully deployed course
- Newsnight covered the Livingstone-Hope NextGen report, which emphasises re-focus on school computing
- David Cameron told the BBC "we're not doing enough to teach the next generation of programmers", and said that the education system needs to change.

Then of course, at BETT in January this year, Michael Gove asked schools to refocus on computer science and programming from Sept 2012 instead of teaching the National Curriculum Programme of Study for ICT. He didn't mince his words, either: "Just at the time when technology is bursting with potential – teachers, professionals, employers, universities, parents and pupils are all telling us the same thing. ICT in schools is a mess."

What's the big deal?

We can all see that computing devices are becoming more common – from home PCs and game systems to smart phones. We have never had such easy access to computing devices and yet the proportion of people using them for creative programming is fewer now than when I grew up in the ZX Spectrum age. Part of the reason for this is that programming environments have got more complex as the industry has matured, and part is that modern devices are built for ease of consumption rather than ease of creativity. After all, consumption leads to the data traffic and publishing licencing revenues that subsidise the devices in the first place to make them affordable. Contrast an iPhone with its AppStore to the ZX Spectrum, which just showed "ok" on the screen when it booted – from there it was down to you as the user to do something. Even loading a game from a tape required one line of code. At a simplistic level we've now got a generation of consumers of technology who know little about how it works. So why is this a problem?

> "MODERN DEVICES ARE BUILT FOR EASE OF CONSUMPTION RATHER THAN EASE OF CREATIVITY..."

The first reason is about lighting the computer science spark and not blowing it out. An enormous part of the UK economy is high-tech. Only a small proportion of children will go on to work in this market, but how many children who would thrive in this industry never even take a look? Turned off by lessons orientated towards the use of IT (as much of ICT has become) we actually present a barrier to them going down this route. Don't get me wrong, I do think children need the sort of skills that ICT teaches, but I think

"We have never had such easy access to computing devices and yet the proportion of people using them for creative programming is fewer now than when I grew up in the ZX Spectrum age."

this can be done quicker, younger and in parallel to more creative and technical teaching. I strongly believe we need universal exposure to programming before taking GCSE options, following which a more technical route through GCSEs and A-Levels to technical degrees must be provided. Let's see in which children the spark lights, and let's not blow it out.

The second reason is the wider benefits computational thinking brings. Nobody just sits at a keyboard and programs. A program is an instruction-by-instruction representation of a thought through approach to solve a problem. Teaching more problem solving and providing tools like programming environments where these solutions can be tested and refined will bring better problem solving skills needed for other courses such as mathematics, the sciences, and even philosophy/PSHE.

The third reason I would offer, as an employer and MD of an IT consultancy, is that we generally need more school leavers with higher technical skills that can be used in industry. It's desperately sad that we have a million unemployed young people in this country, yet we still use technical resources flown in from other countries. More technically aware school leavers, who don't want to go to university, would be ideal to take on as apprentices in areas such as software testing.

What can teachers do?

As Michael Gove said at BETT in January this year "as well as choosing what to study, schools can also choose how." If you interpret this as empowering teachers of a discipline to rapidly evolve that discipline in a more skilled, specialist and technical direction whilst at the same time choosing what to teach and how to teach it then this sounds like a scary prospect!

However, there is lots of help at hand. The Computing At School group is a good start, especially through the regional hub meetings their members organise. I have attended quite a few and they are genuinely a place where teachers with all backgrounds and experience gather to discuss what they are doing, what's working well, and what's not working so well. The annual conference and summer schools also provide a way to get more in depth insight into how computing courses are being run, as well as being opportunities for teachers to gain more skills themselves.

61%

DECLINE IN THE NUMBER OF PUPILS TAKING A GCSE IN ICT BETWEEN 2007 AND 2011

There are also lots of free online resources, including YOUSRC ('you source'), the facility that my business provides, which can enable teachers to gain skills in computer science and then teach them to their students. Social media sources like Twitter have many people contributing ideas in this area, the #CSin2012 hash-tag being a good place to start. Finally, there are an increasing number of events such as "Hack to the Future", held in February 2012 in Preston; Young Rewired State events around the country; Technocamps; and others – all of which allow teachers to increase their skills and involve their students.

My personal advice would be to join Computing At School (it's free) so you get a feed of ideas from other teachers. Then find some time to investigate some of the free resources that are available. Use them to increase your skills whilst thinking through how they would work for you as a teacher, and on your school's IT equipment and network. What works for you is likely to be a specific approach taught in your own individual way, so take time to find what that might be. This ever more urgent evolution should be an opportunity for you to develop new skills and teach new things... something that's always as exciting as it is scary.

PAUL CLARKE IS THE DRIVING FORCE BEHIND YOUSRC ('YOU SOURCE'); A FREE, WEB-BASED LEARN-TO-PROGRAM RESOURCE FOR SCHOOLS. YOUSRC.COM